AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-18. (Canceled)

19. (Currently Amended) A scheduler comprising:

a memory device to store a table, the table to have a total number of N slots to store entries, with L of those slots already containing entries,

wherein N is defined as an integer representing the total number of slots in the table, N further being expressible as a power of two with an integer exponent x, and

wherein L is defined as an integer representing a number of slots that already contain entries; and

~~a plurality of slots equal to two to the power x for entries, the number of the plurality of slots represented by N, and to have a number of entries already entered into the slots represented by L~~

a processor coupled to the memory device, the processor to derive an integer index value [[L']] for a component of a sub-stream, the index value indicating a slot of the table into which an entry associated with the component of the sub-stream is to be placed, wherein the index value is derived by swapping x bits of a binary representation of the number of entries L already in the table.

20. (Currently Amended) The scheduler of claim 19, wherein the entry associated with a sub-stream is a first entry of a series of consecutive entries associated with consecutive components of the sub-stream, and wherein the processor is further to derive an index value for

each of the consecutive entries following the first entry by adding a progressively incremented integer multiple, starting with a multiple of one, ~~of a step associated with the sub-stream,~~ to [[L']] the index value for each consecutive component, respectively.

21. (Currently Amended) The scheduler of claim 19, wherein the processor is to derive the index value [[L']] by exchanging x bits of the binary representation of L around an axis located at one of (i) a center bit of the x bits of the binary representation of L when x is an odd number [[and]] or (ii) between two center bits of the x bits of the binary representation of L when x is an even number.

22. (Canceled)

23. (Original) The scheduler of claim 19, wherein the sub-stream includes constant bandwidth traffic associated with a port coupled to the processor.

24. (Original) The scheduler of claim 19, wherein the processor is further to place an identifier in the index that points to a component of the sub-stream stored in a second table.

25. (Currently Amended) An ATM switch comprising:

a port coupled to a network and receiving a sub-stream therefrom, the sub-stream having a plurality of consecutive components;

a memory device to store a table, the table to have a total number of N slots to store entries, with L of those slots already containing entries,

wherein N is defined as an integer representing the total number of slots in the table, N further being expressible as a power of two with an integer exponent x, and

wherein L is defined as an integer representing a number of slots that already contain entries; and

~~plurality of slots equal to two to the power x for entries, the number of the plurality of slots represented by N, and to have a number of entries already entered into the slots represented by L; and~~

a processor coupled to the memory device, the processor to derive an integer index value [[L']] for a component of a sub-stream, the index value indicating a slot of the table into which an entry associated with the component of the sub-stream is to be placed, wherein the index value is derived by swapping x bits of a binary representation of the number of entries L already in the table.

26. (Canceled)

27. (Original) The ATM switch of claim 25, wherein the sub-stream includes constant bandwidth traffic associated with the port.

28. (Original) The ATM switch of claim 25, wherein the processor is further to place an identifier in the index that points to a component of the sub-stream stored in a second table.

29-30. (Canceled)

31. (New) A method, comprising:

establishing a data structure in a memory unit, the data structure having a total number of N slots for entries,

wherein N is defined as an integer representing the total number of slots in the data structure, N further being expressible as a power of two with an integer exponent x;

storing entries into L slots of the data structure,

wherein L is defined as an integer representing a number of slots that already contain entries;

swapping x bits of a binary representation of L to produce an index value; and

storing a new entry into the data structure, the new entry being stored into an entry represented by the index value produced by said swapping.

32. (New) The method of claim 31, wherein said swapping includes exchanging the x bits of the binary representation of L around an axis located at one of (i) a center at a middle bit of the x bits of the binary representation of L when x is an odd number or (ii) a center between two middle bits of the x bits of the binary representation of L when x is an even number.

33. (New) The method of claim 32, wherein the middle bit is a bit of the x bits such that there is an equal number of the x bits on either side of the middle bit.

34. (New) The method of claim 32, wherein the center between two middle bits is a position such that an equal number of the x bits lie on either side.

35. (New) The method of claim 31, wherein the x bits range from two to the zero power to two to the x minus one power of the binary representation of L.

36. (New) The method of claim 31, wherein said storing includes entering an identifier that serves as a pointer to a first component of a sub-stream in a slot corresponding to the value.

37. (New) The method of claim 31, wherein the data structure includes a table in a scheduler.

38. (New) The method of claim 37, wherein the table is a port shaper table.

39. (New) The method of claim 31, wherein the entries are associated with components of at least one existing sub-stream and the new entry is associated with a first component of a new sub-stream.

40. (New) The method of claim 39, wherein the new sub-stream has a series of consecutive components and the new entry is associated with the first component of the new sub-stream, the method further comprising deriving an index value for each consecutive component of the new sub-stream following the first component by adding a consecutive integer multiple of an integer value to the value obtained by swapping to identify a slot in the table for each consecutive entry of the new sub-stream.

41. (New) The method of claim 40, wherein the new sub-stream includes constant bandwidth traffic associated with a particular port on a network device.

42. (New) The method of claim 41, wherein the traffic includes real-time data traffic.

43. (New) The method of claim 39, wherein the new sub-stream has the largest number of components for entry into the data structure of a plurality of new sub-streams.

44. (New) The method of claim 39, wherein the first component of the new sub-stream includes an ATM cell.

45. (New) The method of claim 39, wherein the first component of the new sub-stream is associated with a particular port on a network device.

46. (New) The method of claim 31, further comprising:

transmitting information through a communication port in accordance with entries stored in the data structure.

47. (New) A computer-readable medium having stored thereon instructions which, when executed, cause a processor to perform a method, the method comprising:

establishing a data structure in a memory unit, the data structure having a total number of N slots for entries,

wherein N is defined as an integer representing the total number of slots in the data structure, N further being expressible as a power of two with an integer exponent x;

storing entries into L slots of the data structure,

wherein L is defined as an integer representing a number of slots that already contain entries;

swapping x bits of a binary representation of L to produce an index value; and

storing a new entry into the data structure, the new entry being stored into an entry represented by the index value produced by said swapping.